



# Relatório

TRABALHO PRÁTICO | ARQUITETURA DE COMPUTADORES I

SOULEYMANE SAAD – Nº49687 | RODRIGO MARQUES – Nº52183

## Conteúdo

<b>Introdução .....</b>	<b>2</b>
<b>Desenvolvimento.....</b>	<b>3</b>
<b>Conclusão.....</b>	<b>6</b>

# Introdução

Neste trabalho foi-nos pedido que desenvolvêssemos um conjunto de funções em assembly RISC-V com a finalidade de obter os contornos de imagens a cores.

O trabalho deverá receber uma imagem em cores, ler a respetiva imagem para um buffer com o tamanho indicado e multiplicado por 3 com o objetivo guardar os pixéis de RGB, aplicar o algoritmo RGB to Gray para obter o buffer a preto e branco e em formato de Gray, utilizar esse buffer a aplicar o algoritmo convolution com o propósito de obter a intensidade dos tons de cinzento do respetivo buffer, após aplicada essa função é aplicada o contour para obter os contornos da imagem usando os dois buffers sobel horizontal e vertical respetivamente e por fim guardar o buffer do contorno em um ficheiro.

Para este trabalho, foi usado a imagem Lenna baseada pelo enunciado dado em formato .jpg e nas dimensões 512x512.

## Desenvolvimento

Foram criados 5 buffer com as dimensões da imagem (512x512), exceto o Buffer que leva as dimensões da imagem mais as 3 posições do RGB (512x512x3).

No main é carregada a imagem original para de seguida serem efetuadas as chamadas das funções.

Funções:

### 1. read\_rgb\_image

- Esta função tem o propósito de ler os pixéis da imagem e copia esses pixéis para um buffer. A função guarda o “file descriptor” no endereço a0 e coloca no a1 o endereço do buffer (Buffer) onde clona os pixéis da imagem para o buffer. No a2 é colocado a dimensão da imagem mais os 3 pixéis do RGB em cada posição do buffer.

### 2. write\_gray\_image

- Esta função tem o propósito de escrever para o ficheiro da imagem (tipo Gray) a partir de um buffer. A função recebe como argumentos o nome de um ficheiro, um buffer com a imagem e o comprimento do buffer.

### 3. rgb\_to\_gray

- converte uma imagem a cores RGB para uma imagem em tons de cinzento GRAY. A função recebe como argumentos um buffer com a imagem RGB e um buffer onde deve ser colocada a imagem em formato GRAY. A função assume internamente que a imagem tem dimensão 512x512.
- Primeiro ele lê o pixel do R e multiplica com a constante indicada pela formula dada pelo enunciado e de seguida faz o mesmo procedimento com os outros pixels. Depois da leitura dos 3 pixels e multiplicados pelas respectivas constantes, é dividida por 100 com o objetivo obter os números decimais indicadas pela formula e por fim é guardada essa conta no a1 onde contem o endereço do Buffer (Buffer2).

### 4. convolution

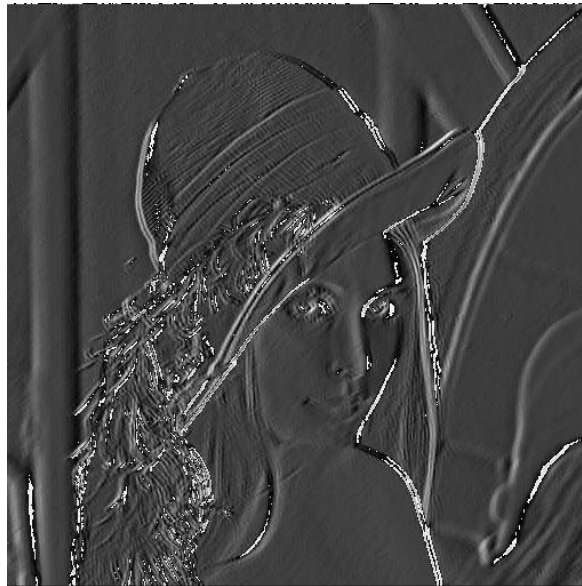
- Calcula a convolução de uma imagem com o operador Sobel (horizontal e vertical, pela respectiva ordem). A função recebe como argumentos um buffer com a matriz A, um buffer com um dos operadores Sobel e um buffer que vai conter a imagem filtrada B. A função pode assumir internamente uma dimensão de imagem fixa ou ter a dimensão como parâmetros.

## 5. contour

- calcula a imagem final combinando as duas imagens convolvidas. A função tem como argumentos dois buffers com as imagens a combinar e um buffer que vai conter o resultado. A função pode assumir internamente uma dimensão de imagem fixa ou ter a dimensão como parâmetros.

## Conclusão

Na realização deste trabalho não foram atingidos todos os objetivos. Foi possível usar os operadores Sobel, porém faltaram alguns pixels na imagem (Figura 1), o que torna o procedimento de obter os contornos a não estar 100% capaz e com isso a imagem final dá uma tela preta. Tentaremos o fazer até ao dia de apresentação deste mesmo projeto.



*Figura 1 - Sobel Horizontal*

Contudo, o nosso programa consegue abrir o ficheiro e transformar a imagem a preto e branco e converte-a em formato gray.

A realização do programa permitiu-nos um conhecimento mais aprofundado sobre a utilização do assembly.